Dumb software agents on an experimental asset market

Jens Großklags[*], Carsten Schmidt[+], Jonathan Siegel[*]

Institute of Information Systems[*]

Institute for Economic Theory III[+]

Humboldt-Universität zu Berlin

Spandauer Str. 1

D-10178 Berlin, Germany

{jensg,cschmidt,jsiegel}@wiwi.hu-berlin.de

September 24, 2000

Abstract

We have analyzed the impact of agents and their trading strategies on an experimental electronic market. Therefore, we added an XML-interface to an existing electronic market and implemented artificial agents which acted as elements of disturbance in the trading process. These artificial traders applied simple and constant strategies which may sometimes appear to be „rational" or random to the eyes of other traders. We then stepped back and recorded the reaction of the electronic market.

Keywords: *double auction, negotiation, XML*

We would like to thank Jan Hansen, Martin Strobel, the participants of the „Experimental Electronic Markets" seminar at Humboldt University Berlin, and an anonymous referee, who read an extended abstract of this paper, for helpful comments. Support from the German Research Society (DFG grant nos. SFB 373/C5 and GRK 316) is gratefully acknowledged.

1. Introduction

As electronic media and computer networks made their appearance in the world of finance, traders began to use software programs that could help them exploit the possibility of market gains that are generally forgone by humans. This is either due to the rapidity with which the transaction must be executed or to the great amount of data that has to be processed in order to perform these transactions. Such software programs - also known as „agents“ - are being utilized for different transactions and, consequently, differ greatly in complexity, ranging from the very simplest to complicated programs which require statistical analysis.

One of the possible future applications of agents is the field of electronic commerce. In this particular context a commercial interaction is composed of at least three distinct steps. First, there is a “searching moment” in which buyers and sellers look and find one another; then the two parties negotiate the terms of exchange; finally there is a settlement and the exchange of the good (Guttmann et al. 1998, Wurman et al. 1998). Aim of this paper is to study the negotiation procedure of artificial agents acting on electronic markets that incorporate a centralized auctioneer process. Although our investigation seems to be more suited for business-to-business electronic commerce, where a “variable-price model” is well established, it may be also applied to the consumer side: in fact there are strong signals in this context - such as online auctions - that the “fixed-price model” may not be the only possible scenario in retail electronic commerce.

To be more specific, the paper analyzes the impact of agents and their trading strategies on an experimental electronic market. For this purpose, we added an XML-interface to an existing electronic market and introduced artificial agents which act as elements of disturbance in the trading process. These artificial traders apply simple and constant strategies which may sometimes appear „rational“ or random to the eyes of other traders. We then stepped back and recorded the reaction of the electronic market.

The research emphasis lies not in finding the agent which applies the best strategy: other papers have dealt with this problem in much greater detail (Rust et al. 1993). Rather the emphasis lies with the “market” and with questions such as: “How does the market react to agents?” Far from being obvious, this question is far more complicated than intuition would prompt to think. There has been a discussion on the lower bound of traders’ intelligence to act similarly to human traders in a market institution. Gode and Sunder (1993), for example, introduce “zero-intelligence” traders that act randomly and, nevertheless, converge to the theoretical equilibrium price within a continuous double auction (CDA) framework, suggesting that price is determined more by market structure rather than by the intelligence of the traders. On the other hand, Cliff and Bruten (1998) present criticism to this point of view, arguing that these results are an artifact of the experimental regime chosen by Gode and Sunder and introduce “zero-intelligence plus” traders, which seem to have some more “intelligence” by employing a modified adaptive version of the random agent.

The paper is organized as follows: In the next Section we discuss the technical implementation of the market-agent interface. Section 3 gives an overview on the set of the implemented trading strategies. Next comes the presentation of the results of the experiment in Section 4. We then defer a discussion of related work to Section 5 in order to be able to better place our work in the context of related efforts. Open research questions for further exploration and concluding remarks are given in Section 6.

2. Market-agent interface

In this Section we first present a short description of the electronic market, whereas the market itself is regarded as a black box. We then focus on the information available to the agents and on the communication between agents and the market.

2.1 Market rules

Our market is an artificial stock market infrastructure that has been used for several sports and election markets.[1] So far, human traders invest their own funds and buy/sell shares, based on their prediction about the outcome of some kind of event. The experiments that have been run with human trader are in the tradition of experimental economics (Smith 1976) in that participants are paid according to their results. The design of the market follows closely the Iowa Electronic Markets, who have pioneered election markets (Forsythe et al. 1992).

This kind of electronic market is always open, i.e. the traders may trade anytime, at any hour, and they are always accessible from anyplace in the world which is connected to the Internet. Traders have an account at the bank and can act on the market by buying and selling assets. The market is a centralized information hub in which all the transactions are explicitly regulated via predefined and fixed rules and the agents interact via the market system. In the following the market rules are explained in brief.

The artificial stock market uses the continuous double auction (CDA), i.e. an auction in which sellers and buyers may submit bids and asks simultaneously and asynchronously: sellers and buyers are free to accept bids or asks at any time. The CDA is very popular among financial markets, both real and virtual, and is thought to have the remarkable quality of being fast and efficient (Friedman 1984, 1993). In contrast to markets where the emission of shares is organized by an initial public offering, the emission of shares on this market is implemented via a so called bundle mechanism. The bundle consists of a standardized portfolio of one piece of each share at a fixed price: this can be bought from or sold to the bank at any time and any quantity. The market foresees three valid operations: (1) posting market orders (bids implement buy orders and asks sell orders), (2) deleting own market orders, and (3) buying/selling bundles at the bank.

The market actually implements an American futures market, where shares can be traded on some kind of event. The outcome of the event determines, depending on the market rules, the payoff of the different shares. For the experiment in Section 4 we have used a winner takes it all (WTA) payoff scheme, where at the end of the market one share pays off the price of a bundle, and all other shares pay off zero. Compared to vote share payoffs, where each share pays off a fixed percentage of the total bundle, WTA markets are considered a risky environment: when the agent does not have the winning share in his portfolio, he will loose all his invested money. We used a WTA market for the experiment, because they have the nice property, that share prices relate directly to the probability of a share paying off, and human trader often prefer them over vote share markets.

2.2 Information

In order to apply a strategy the agents require information about the market. One of the design goals of the market-agent interface was to give the same information to human and to artificial agents. For this purpose we analyzed the information available to human traders and categorized the information into the following macro-categories:

---

[1] Soccer world championships 1998, http://wm.wiwi.hu-berlin.de, State elections Berlin 1999, http://www.wahlboerse-berlin.de, European soccer championships 2000, http://www.ribaldo7.de.

(1) rule information
   - basic bundle composition
   - bundle price (bought/sold to the bank)
   - payoff rules at the end of the market
   - start and stop time of the market
(2) public market information
   - last traded price of each share
   - current bids and asks of each share
   - order book: the top 5 bids and asks including prices and quantities of each share
   - server time
(3) personal information
   - portfolio of the trader
     - type of shares
     - quantity of shares
   - liquid money
(4) transaction status information
   - validity of transaction
   - success of transaction

Out of the above categories we designed the framework for the market-agent interface. The eXtensible Markup Language (XML) was chosen as high level data exchange format. Two Document Type Definitions (DTD) incorporating the above information categories were developed: the first consists of rule, market, and personal information; the second returns the result status when an agent is performing one of the three legal market transactions (Fig. 1).

DTD 1:

```
<?xml version='1.0' encoding='us-ascii'?>
<!--    DTD for agent-experiments on electronic markets -->
<!ELEMENT MARKET (PUBLIC,PRIVATE?)>
<!ELEMENT  PUBLIC  (MARKETID,  BUNDLEPRICE,  NO_OF_STOCKS,  DIGITS,  OPENINGHOURS,
STOCK+)>
<!ELEMENT MARKETID (#PCDATA)>
<!ELEMENT BUNDLEPRICE (#PCDATA)>
<!ELEMENT NO_OF_STOCKS (#PCDATA)>
<!ELEMENT DIGITS (#PCDATA)>
<!ELEMENT OPENINGHOURS (BEGIN, END, NOW)>
<!ELEMENT STOCK (ID, LASTPRICE, OFFER*)>
<!ELEMENT OFFER (BID*, ASK*)>
<!ELEMENT BID (PRICE, QUANTITY)>
<!ELEMENT PRICE (#PCDATA)>
<!ELEMENT QUANTITY (#PCDATA)>
<!ELEMENT ASK (PRICE, QUANTITY)>

<!ELEMENT PRIVATE (USERNAME, BALANCE, AVAILABLE, STOCK+)>
<!ELEMENT PRIVATE.USERNAME (#PCDATA)>
<!ELEMENT PRIVATE.BALANCE (#PCDATA)>
<!ELEMENT PRIVATE.AVAILABLE (#PCDATA)>
<!ELEMENT PRIVATE.STOCK (ID, PORTFOLIO, OFFER*)>
<!ELEMENT PRIVATE.ID (#PCDATA)>
<!ELEMENT PRIVATE.PORTFOLIO (#PCDATA)>
<!ELEMENT PRIVATE.OFFER (BID*, ASK*)>
<!ELEMENT PRIVATE.BID (PRICE, QUANTITY, EXPIRATION_TIME, CREATION_TIME)>
<!ELEMENT PRIVATE.PRICE (#PCDATA)>
<!ELEMENT PRIVATE.QUANTITY (#PCDATA)>
<!ELEMENT PRIVATE.EXPIRATION_TIME (#PCDATA)>
<!ELEMENT PRIVATE.CREATION_TIME (#PCDATA)>
<!ELEMENT PRIVATE.ASK (PRICE, QUANTITY, EXPIRATION_TIME, CREATION_TIME)>
```

DTD 2:

```
<?xml version='1.0' encoding='us-ascii'?>
<!--    DTD for agent-experiments on electronic markets,
        result codes of transactions -->
<!ELEMENT RESULT (CODE,MESSAGETEXT)>
<!ELEMENT CODE (#PCDATA)>
<!ELEMENT MESSAGETEXT (#PCDATA)>
```

Figure 1: XML DTD‘s for the market-agent interface.

2.3 Agent-Market Communication

The implemented communication between the agent and the market may be broken down into several steps (Figure 2): (1) the agent makes a request to the electronic market via HTTP, (2) the electronic market replies by supplying the relevant market information as an XML document, (3) the agent applies its strategy to the data and decides whether it should perform one of the three legal actions (see Subsection 2.1) or do nothing, (4) the agent sends the information regarding its decision to the electronic market via HTTP and (5) receives an XML document with the status of the transaction.

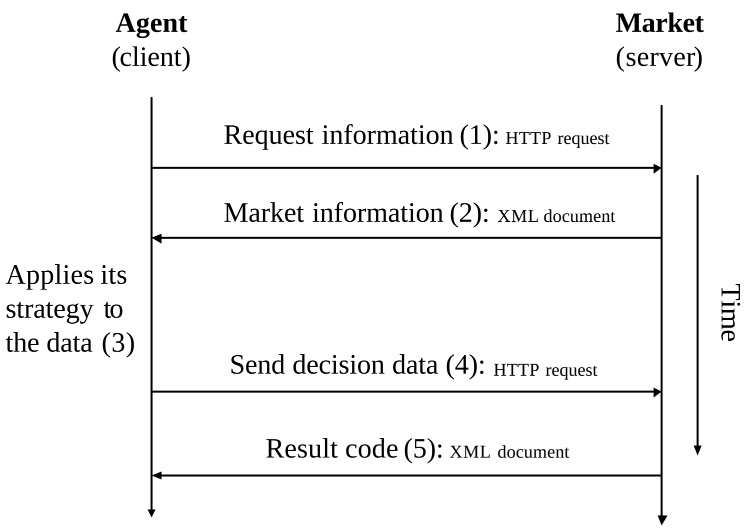

Figure 2: Time sequence diagram: communication between agent and market

In the following, the communication steps are described in more detail. Hereby we also present the URLs for the complete set of available transactions and the resulting XML-files. The data exchange between the agents and the market happens, as said before, via the Internet. Low level exchange protocols are therefore the TCP and IP protocols and the high level data exchange format is the HTTP protocol.

In step (1) the agent performs a HTTP-request to obtain private and public information:

```
http://market.wiwi.hu-
berlin.de:7000/xml/market.xml?username=myusername&password=mypassword
```

The market responds in step (2) with a resulting XML-file (Figure 3). Note that private information is only supplied when a valid username/password combination is used. Without valid identification the XML-file will only contain rule and public market information.

```
<?xml version="1.0"?>
<MARKET>
   <PUBLIC>
      <MARKETID>0</MARKETID>
      <BUNDLEPRICE>1000</BUNDLEPRICE>
      <NO_OF_STOCKS>16</NO_OF_STOCKS>
      <DIGITS>0</DIGITS>
      <OPENINGHOURS>
         <BEGIN>960188969</BEGIN>
         <END>982569800</END>
         <NOW>962645225</NOW>
      </OPENINGHOURS>
      <STOCK>
         <ID>0</ID>
         <LASTPRICE></LASTPRICE>
         <OFFER>
            <BID>
               <PRICE>100</PRICE>
               <QUANTITY>1</QUANTITY>
            </BID>
            <ASK>
               <PRICE>500</PRICE>
               <QUANTITY>10</QUANTITY>
            </ASK> [...code follows...]
         </OFFER>
      </STOCK>
      <STOCK>
         <ID>1</ID> [...code follows...]
      </STOCK>
   </PUBLIC>
   <PRIVATE>
   <PRIVATE.USERNAME>jensg</PRIVATE.USERNAME>
   <PRIVATE.BALANCE>495000</PRIVATE.BALANCE>
   <PRIVATE.AVAILABLE>410000</PRIVATE.AVAILABLE>
   <PRIVATE.STOCK>
      <PRIVATE.ID>0</PRIVATE.ID>
      <PRIVATE.PORTFOLIO>5</PRIVATE.PORTFOLIO>
      <PRIVATE.OFFER>
         <PRIVATE.BID>
            <PRIVATE.PRICE>155</PRIVATE.PRICE>
            <PRIVATE.QUANTITY>15</PRIVATE.QUANTITY>
            <PRIVATE.EXPIRATION_TIME>982569800</PRIVATE.EXPIRATION_TIME>
            <PRIVATE.CREATION_TIME>962640181</PRIVATE.CREATION_TIME>
         </PRIVATE.BID>
         <PRIVATE.BID>
            <PRIVATE.PRICE>135</PRIVATE.PRICE>
            <PRIVATE.QUANTITY>23</PRIVATE.QUANTITY>
            <PRIVATE.EXPIRATION_TIME>982569800</PRIVATE.EXPIRATION_TIME>
            <PRIVATE.CREATION_TIME>962640185</PRIVATE.CREATION_TIME>
         </PRIVATE.BID>
         <PRIVATE.BID>
            <PRIVATE.PRICE>75</PRIVATE.PRICE>
            <PRIVATE.QUANTITY>15</PRIVATE.QUANTITY>
            <PRIVATE.EXPIRATION_TIME>982569800</PRIVATE.EXPIRATION_TIME>
            <PRIVATE.CREATION_TIME>962640061</PRIVATE.CREATION_TIME>
         </PRIVATE.BID>
         </PRIVATE.OFFER> [...code follows...]
      </PRIVATE.OFFER>
   </PRIVATE.STOCK>
   </PRIVATE>
</MARKET>
```

Figure 3: XML interface definition for the private and public information

Based on the information contained in the XML-file the agent applies its strategy to the data in step (3), which is outlined in more detail in the next section. In step (4) the agent has the option to initiate three valid market transactions:

posting bids and asks on the market:

```
http://market.wiwi.hu-
berlin.de:7000/xml/trade.xml?notationID=0&marketID=0&transaction=b&price=10&q
uantity=5&expiration_time=&username=myusername&password=mypassword
```

deleting its own bids or asks on the market:

```
http://market.wiwi.hu-
berlin.de:7000/xml/delete.xml?notationID=0&marketID=0&transaction=b&price=10&
quantity=5&expiration_time=&username=myusername&password=mypassword
```

buying bundles form or sell bundles to the bank:

```
http://market.wiwi.hu-
berlin.de:7000/xml/bundle.xml?marketID=0&transaction=b&quantity&username=myus
ername&password=mypassword
```

In step (5) the market responds with an XML-file regarding the status of the transaction (Fig. 4). Now the agents continue with step (1) in order to get new market information.

```
<?xml version="1.0"?>
<RESULT>
<CODE>0</CODE>
<MESSAGETEXT>
The market is closed!
</MESSAGETEXT>
</RESULT>
```

```
<?xml version="1.0"?>
<RESULT>
<CODE>1</CODE>
<MESSAGETEXT>
Transaction   successful!   Bought   4
bundles from the bank!
</MESSAGETEXT>
</RESULT>
```

Figure 4: XML interface definition for the three market transactions: market was closed (left), and market was operating (right).

3. Agents and strategies

Considering the game theoretical point of view we can define strategy as a player's complete plan of action for all possible occurrences in the game. From the computer science point of view, strategy is the execution of a fixed algorithm. The two definitions are, of course, not mutually exclusive and indeed we need both of them.[2] However, because we will describe the actions of the agents we have implemented, we will stick to the latter interpretation of the strategies. We have chosen the sample of strategies on behalf of basic and pure strategies which have been observed in artificial stock markets with human traders. This compilation may not be complete nor is it designed to find out a successful strategy. Here we first describe the general idea behind each strategy, then the reason for which we have created such a strategy (the "real world analogy") and finally we describe, in words, the algorithm (the

[2] Different would be the case of a market that does not use software agents. In this case the game theoretical interpretation would be sufficient

instance). The agents and their strategies have been implemented as a stand-alone Java-client (JavaSoft 1997), that can be used on any standard personal computer.[3]

3.1 The fundamentalist (patriot style)

<u>Strategy description</u>: A "fundamental strategy" is one in which a trader buys a share and waits for the final event to happen, trusting his/her analysis of the fundamental data regarding the entity (company, soccer team, etc.) whose share is bought.

<u>Real world analogy</u>: Fundamentalist actions are quite common among traders in real world markets. Traders, for example, buy shares relying on financial information regarding the firm (fundamental data) and keep them regardless of short-run price fluctuations of the shares: the fundamentalist strategy may be seen as a sort of patriotic behavior, in which the trader buys the shares of his home team.

<u>Instances</u>: We implemented one *fundamentalist* agent. It basically concentrates its efforts only on one entity and buys the shares only of that entity. For simplification this agents posts only bids: the price at which the shares are bought is the last paid price on the market or, if possibly, lower.

3.2 The follower

<u>Strategy description</u>: A "follower strategy" is one in which a trader echoes the actions of the other subjects of the market based on a pre-defined signal. In our case, the agent reacts to the price of a share (the signal), buying when the price of the share is high or low. The underlying assumption is that the price of a share is a signal regarding the quality of the entity (e.g. a firm): if the share's price is high, this means that many traders are asking for this share and therefore we may presume that the share is very valuable. On the other hand, if the price of the share is very low, we may come to the conclusion that the entity whose share is traded is not very valuable.

<u>*Real world analogy:*</u> Follower actions are quite common. Many traders rely only on the trend of the price of the share, sometimes acting as if the share had its own life and did not relate to a real world entity. Chart analysis, analysis based on historical data of the share's price, is nowadays a "science" to which many traders relate.

<u>Instances</u>: We implemented two follower agents:

1. The *follower high* buys the share with the highest price and does it as long as there is enough money. To put it differently, this strategy prompts to follow the mass: if the expectations about the final value of the share are high, everyone will be willing to buy the share. Thus, its price will be very high. This agent "adapts" its expectations to the others.
2. The *follower low* buys the share with the lowest price and does it as long as there is enough money. It is basically the opposite of the first one. The trader buys the shares which are discarded by all other traders. The price of the share, again, reflects the expectation about the final value of the share: the entity whose share is bought is thought of being a sure looser. With a high probability the trader is going to incur a loss. Nevertheless, if , at the final event, the shares bought are the winning ones, the agents will gain considerably.

[3] The source code is available at http://www.wiwi.hu-berlin.de/~jensg/etrade. The following requirements are needed in order to run the agents: JDK 1.1.8 or later (including the Java 2 SDK), JAXP - JAVA API for XML Parsing, http://java.sun.com/xml.

3.3 The stochastic

Strategy description: A "stochastic strategy" is one in which a trader buys shares at a random price. In our case the agent does not choose a completely random price in the set of all possible numbers, rather a price which floats in a fixed range or in a range around a given value. We have implemented two kind of variations, one that uses the market rule information and one that uses the public market information to calculate the given value.

Real world analogy: Stochastic actions are difficult to be proven in a real stock market: even if we could find a trader who candidly admitted that he/she traded without knowing what he/she was doing, there is always a rationale behind a trading action. Nevertheless traders have different degrees of information, and traders who dispose of more information may appear to act irrationally in the market to the eyes of poorly informed traders. A typical example may be the action of an "inside trader" that handles shares in a rational manner, but my appear "stochastic" to the eyes of non informed traders.

Instances: We planned four stochastic agents. All of them continuously and randomly chose the shares they bought and sold.

1. The *stochastic* generates its stochastic price bid/ask based on the following formula: "*the stochastic price is an equally distributed value which lies between 0 and 1.5*bundle-price/number of shares*"
2. The *static stochastic* generates its stochastic price bid/ask based on a fixed predetermined parameter, so the prices of its orders will be randomly distributed around a given value. The formula states "*the stochastic price lies in the* range of ± 100*RV around a given value", where the random value (RV) is normally distributed between –1 and +1.
3. The *dynamic stochastic* generates its stochastic price bids/asks based on a constantly changing parameter, the last price on the market. The formula states: "*the stochastic price lies in the range of ± 100*RV around the last paid price*", where RV is a normally distributed random value between –1 and +1. If we may say so, the dynamic stochastic agent has a higher degree of adaptation to the sub events of the market than the static stochastic.
4. The *demand function stochastic* builds its strategy on the *dynamic stochastic* agent, but adds a peculiarity: it follows the price of a share and modulates the quantity accordingly. Formally speaking, the agent has a fixed value for the price of a share (threshold value) under which it buys the share with increasing quantities as the price of the share declines, and over which it sells the share with increasing quantities as the price increases. In more simple words: (when acting as a seller) the more the share is in demand, the higher is the price and the more this agent sells that share; on the other hand (when acting as a buyer) the more the share is sold, the lower is the price and the more this agent buys of that share. It basically follows the demand or the offer curve.

3.4 The arbitrageur

Strategy description: An "arbitrageur strategy" tries to earn a profit without taking a risk. This is implemented by buying a standardized portfolio from the market and selling it back to the bank or the other way around. Differently stated, the agent exploits the difference between market price and bank price for the same bundle. This will lead to sure profit when the agent can perform the complete set of transactions.

Real world analogy This is a more elegant strategy and probably the one which bears the greatest resemblance to real trade strategies. Arbitrage trading happens daily in the markets. These actions may be performed by expert traders who can simultaneously follow different markets, but, since these actions are quite tedious and need constant attention, they are generally performed by software agents.

Instances: For the *arbitrageur* strategy we planned one agent. The agent considers the shares which form the bundle **X**, the bundle that the bank is always ready to buy and sell at any time at a fixed price. Then it looks for the outstanding bids/asks of the single shares forming the bundle. If the overall ask of the bundle is lower than the bank price the agent will buy the bundle from the market and afterwards resell it to the bank. On the contrary, if the best buy offers (bid) on the market sum up to a higher bundle price than the price of the bank the agent will buy the bundle from the bank and resell it to the market.

3.5 The speculator

Strategy description: This agent applies a similar strategy to the arbitrageur, though with a small but relevant difference: it does not aim for a sure win by exploiting the difference between market price and bank price, rather it “raids on the wave of speculation”, pushing the willingness to pay of the other traders. To put it differently, it gives them the shares they want gambling on the fact that in a situation of collective excitement, traders will be ready to pay a little bit more in order to possess the “hot” share.

Real world analogy: Speculative bubbles may be our best examples. People - both experts and non experts - buy shares because of “voices” or because in recent times the share’s value has risen tremendously (possibly because others before bought them based on the same reasoning). The speculator hopes to make money not from the “dividend” of the share, but from the difference between buying price and selling price, trusting that the price of the share will continue to rise even more after the share was bought.

Instances We planned one *speculator* agent. It looks at the market price of the shares or, better said, of the bundle of shares: if the market price of the bundle is overvalued, that is, if the sum of the ask is greater than the bundle price at the bank , the agent buys the bundle from the bank and tries to resell it to the market at the overvalued price plus a mark-up. The formula states: *“buy stock form the bank and resell it in the market at the last-price + RV* given value”*, where the random value (RV) is between 0 and +1 the right half of a normal distribution. The “given value” is a parameter which gives the higher cap on the mark-up or, in other words, how much more the agent may ask at maximum.

4. Experimental results

To test the market-agent interface, we conducted a simple experiment where the market design was taken from the previously running Euro2000 market: traders could buy and sell shares of 16 countries participating in the European soccer championships with a payoff of 1.000 ECU (Experimental Currency Unit) for the European Champion and a payoff of 0 ECU for all other 15 teams. The agent experiment used the same rules, yet there were some changes: the a priori probability for each share being the winner was 1/16, the market opened only for one hour, and during that time only market information was available to the agents. Each of the 11 agents started with an initial endowment of 500.000 ECU and without any shares. The exact parameters of the agents’ strategies are described in Table 1. To give all agents an equal opportunity to trade on the market each instance of an agent was started on the same computer consuming the same amount of CPU time and memory.

#	Agent	Parameter	Value	Market transactions	Market-adaptive
1	*Fundamentalist (1)*	ID of share limit price	0 250	market bid	yes
2	*Fundamentalist (2)*	ID of share limit price	5 200	market bid	yes
3	*Follower High*	limit price	500	market bid	yes
4	*Follower Low*			market bid	yes
5	*Pure stochastic*			bank buy/sell, market bid/ask	no
6	*Static stochastic (1)*	mean price	65	bank buy/sell, market bid/ask	no
7	*Static stochastic (2)*	mean price	60	bank buy/sell, market bid/ask	no
8	*Dynamic stochastic*			bank buy/sell, market bid/ask	yes
9	*Demand function stochastic*			bank buy/sell, market bid/ask	yes
10	*Arbitrageur*			bank buy/sell, market bid/ask	yes
11	*Speculator*	max. surplus	30	bank buy, market ask	yes

Table 1: Agents and parameter values

The agents can be described as stable, i.e. all agents traded until the end of the session, independent, i.e. all of them could start and stop independently from others, interface-human-like, i.e. all agents used the same information that the market provides to human traders, and market-adaptive, i.e. the majority (ruling out, for example, the static stochastic agent) made their transaction based on market information. During a one hour trading period the average trading volume was 2,4 Mio ECU and there was about one market transaction, where buy and sell offers were matched, every second.

During a market run, the first traders are the non-adaptive agents: these are the one which start posting offers and trade with each other. At the point where the market is over-/undervalued the *arbitrageur* starts trading. When prices for all shares develop all other market-adaptive agents start to act. A very active trader in selling shares is the *speculator*, whereas the *follower low* is the most active in buying shares. Yet the overall most active traders are the *stochastic* agents, which are doing both, buying and selling on the market. A more background-like strategy can be observed for the *fundamentalists*, which try to buy their specific share at a reasonable price.

#	Agent	No of market transactions: buy	No of market transactions: sell	First trade after X market transactions	No of different shares in portfolio	Yield
1	*Fundamentalist (1)*	209	0	129	1	-40,9%
2	*Fundamentalist (2)*	161	0	88	1	-36,9%
3	*Follower High*	197	0	107	5,3	-45,1%
4	*Follower Low*	622	0	425	13	37,2%
5	*Pure stochastic*	570	961	8	15	11,6%
6	*Static stochastic (1)*	601	452	21	16	34,6%
7	*Static stochastic (2)*	567	474	6	15,6	30,9%
8	*Dynamic stochastic*	222	151	34	16	9,3%
9	*Demand function stochastic*	191	141	32	16	10,2%
10	*Arbitrageur*	40	236	18	14,6	2,2%
11	*Speculator*	0	965	279	16	-13,1%

Table 2: Descriptive agent results, values are the average of 3 market runs.

The time series of the market prices of the 16 shares can be divided in two categories. The shares the *fundamentalist* strategies are buying (0 and 5) on the one hand, and the rest of the shares on the other hand. The private information of the fundamentalist, to buy share 0 up to until 250 ECU and share 5 up to until 200 ECU, results in a concave rise of these share prices, yet the full information is not revealed (Figure 5, top left). The public information, that all shares have a winning probability of 1/16, is revealed in the market by similar prices for the rest of the shares. These shares do not seem to follow a pattern, which may give a hint on a random walk (Figure 5, top right). When we look at information efficiency, Fama (1970) distinguishes between markets with fully revealed private information and not revealed private information. Our market experiment revealed the private information partly, which goes in accordance with empirical market data.

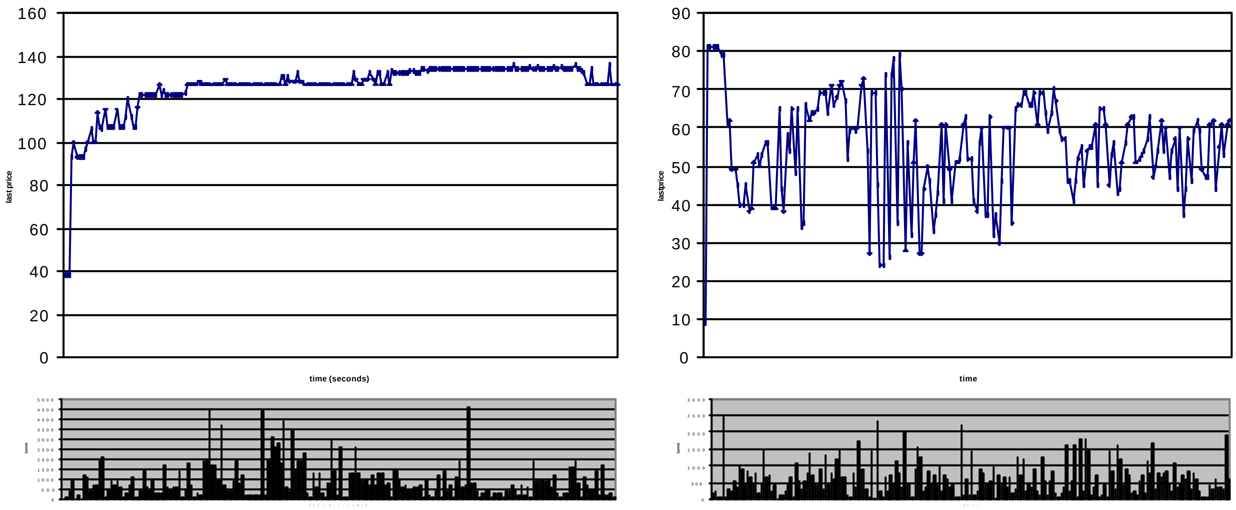

Figure 5: Sample time series of traded prices of share 5 (top left) with the corresponding trade volumes (bottom left) and share 4 (top right and bottom right).

To our surprise the *arbitrageur* only yielded a tiny win on average and even lost money on occasions. It turned out that the agent could often not complete the whole set of trades to employ its arbitrage strategy. This is due to the market mechanism, which does not allow for atomic transaction for a set of trades in order to gain arbitrage. In the case of the *arbitrageur* the result was an unbalanced portfolio, which is equal to engaging in a risk. The strategy may be improved by re-balancing the portfolio.

Given the set of implemented agents and the WTA market rule, the most successful strategy is the *follower low*, who attempts to buy a portfolio of low priced shares. This agent starts his first market transaction late, waiting for the other agents to generate prices. At this point the strategy is very effective in conducting market transactions that implement the successful portfolio (see Table 2). Similar to the most successful strategy described by Rust et. al (1994) the agent can be described as "sit in the background". Yet the *follower low* would not be successful, when the *fundamentalist* is buying the winning share, driving the prices of the share up. Runner up is the *static stochastic* strategy, which also manages to get a diversified portfolio by posting rather constant prices on every share. Not successful have been the specializing *fundamentalists*, who attempt to buy an extremely unbalanced portfolio. These strategies lost all their invested money in picking up a share that did not pay off.

5. Related work

Much work has been done about automated agents in the context of electronic commerce. A good starting point on agent mediated electronic commerce can be found in Guttmann et al. (1998). In the following, we will focus on related work that is concerned about automated negotiation. In particular we will review market based approaches, which provide a market institution and a set of rules to do the negotiation. In our context agents negotiate in a competitive environment, yet there are other approaches in the AI community, such as collaborative agents, that will not be reviewed here.

In the experimental economics community work on strategies has often been done by conducting tournaments (Abreu and Rubinstein 1988, Selten et al. 1997). Rust et al. (1993, 1994) report on the *Santa Fe's tournament*, where researchers where invited to submit software agents that compete on a CDA market against one another. The focus here was to find successful trading strategies out of the submitted set of agents. The most successful strategy in this tournament can be described as rather parasitic: sitting in the background and exploiting the strategies of other traders. In addition, they report about an evolutionary tournament, where the percentage of traders was adjusted in accordance to the success of a strategy over time. Parallel to the tournament there has been a discussion on the lower bound of traders' intelligence to act similarly to human traders in a market institution (Gode and Sunder 1993a&b, Cliff and Bruton 1998). A recent overview on agent-based computational finance can be found in LeBaron (2000).

Applications for multi-agent systems using the CDA mechanism have been developed by the XeroX Palo Alto Research Center (Parc). This includes a system for controlling building environments, where cool air is auctioned in the Parc building (Clearwater and Huberman 1994, Huberman and Clearwater 1995). In this application thermostat-agents buy and sell cool air with the help of a central auctioneering process. Other applications include computational resource allocation (Huberman and Hogg 1995).

The caveat of CDA systems is that they will only be useful for spot markets, where homogenous goods can be described with a single price. Systems that handle goods with multi dimensional properties include *Kasbah* developed by MIT, a system where users can create autonomous agents that buy and sell goods on their behalf (Chavez and Maes 1996, Chavez et al. 1997). This system uses a classified ad metaphor, where agents post their offers to the common blackboard and agents rely on one-to-one bargaining. A market based approach is the *Michigan AuctionBot* where human and artificial agents can engage in online auctions via Internet (Wurman et al.1998). Other systems include the *fishmarket* project (Collins et al. 1998) and *MAGNA*, an integrated approach to an agent-based virtual market that should include all steps of a commercial interaction (Tsvetovatyy et al. 1997).

An introductory text for using economic principles in automated negotiation can be found in Binmore and Vulkan (1999). Varian (1995) gives an introduction to economic mechanism design for computerized agents.

6. Conclusions and Outlook

We have extended an artificial stock market for human traders with an automated agent interface. The use of a market institution results in a simple negotiation protocol that was implemented by using eXtensible Markup Language (XML). In our first experiment we used a set of agents, that implemented pure and simple strategies on the ground of market rules and market information. These agents, some of which show random, others market adaptive behavior, can simulate human-like trading performance in the existence of a market institution.

Further work should compare strategies in different payoff regimes that impose different degrees of risk, e.g. vote share markets compared to winner takes it all markets. The analysis of the strategies should include a larger set of agents, that might be collected in a tournament, where traders might send in their own strategies. In addition, the stability of successful strategies can be tested in an evolutionary experiment, where the population of strategies converges towards the more successful strategies. So far the automated agents only trade on behalf of rule and market information. An extension could be to allow for an additional news source, that provides changing information about the stocks.

The ultimate goal for further experiments is to inductively observe how agents change the strategies of human subjects operating in the electronic market. Now, the market-agent interface gives us the ability to explore this question in a laboratory environment, where human and artificial agents can participate alike.

References

Abreu, D., A. Rubinstein (1988) "The Structure of Nash Equilibrium in Repeated Games with Finite Automata", Econometrica 56(6): 1259-1281.

Binmore, K. , N. Vulkan (1999) "Applying game theory to automated negotiation", Netnomics 1(1): 1-9.

Bray, T., J. Paoli, and C.M. Sperberg-McQueen (eds.) (1998) "Extensible markup language (xml) 1.0", Technical report, W3C.

Chavez, A., D. Dreilinger, R. Guttman, P. Maes (1997) "A Real-Life Experiment in Creating an Agent Marketplace", Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM '97). London, UK.

Chavez, A., P. Maes (1996) "Kasbah: An Agent Marketplace for Buying and Selling Goods", Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology.

Clearwater, S.H., B. A. Huberman (1994) "Thermal Markets for Controlling Building Environments", Energy Engineering 91(3): 26-56.

Cliff, A., J. Bruten (1998) "Less Than Human: Simple adaptive trading agents for CDA markets", Technical Report HP-97-155, Hewlett Packard Research Laboratories.

Collins, J., B. Youngdahl, S. Jamison, B. Mobasher, M. Gini (1998) "A Market Architecture for Multi-Agent Contracting", 2nd Int'l Conf. on Autonomous Agents, Minneapolis: 285-292.

Fama, E.F. (1970) "Efficient capital markets: A review of theory and empirical work" Journal of Finance 25:383-417.

Forsythe, R., F. Nelson, G.R. Neumann, J. Wright (1992) "Anatomy of an Experimental Political Stock Market", AER 82(5): 1142-1161.

Friedman, D. (1984) "On the efficiency of Experimental Double Auction Markets", AER 74(1): 60-72.

Friedman, D. (1993) "The Double Auction Market Institution: A Survey", in Friedman, D., J. Rust (eds.) "The Double Auction Market – Institutions, Theories, and Evidence", Proceedings volume, Santa Fe Institute studies in sciences of complexity: 14, Addison-Wesley.

Glushko, R.J., J.M. Tenenbaum, B. Meltzer (1999) "An XML Framework for Agent-based E-Commerce", Communications of the ACM, 42(3): 106-114.

Gode, D.K., S. Sunder (1993a) "Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality". Journal of Political Economy, 101 (1), 119-137.

Gode, D.K., S. Sunder (1993b) "Lower Bounds for Efficiency of Surplus Extraction in Double Auctions", in Friedman, D., J. Rust (eds.) "The Double Auction Market – Institutions, Theories, and Evidence", Proceedings volume, Santa Fe Institute studies in sciences of complexity: 14, Addison-Wesley.

Guttmann, R.H., A.G. Moukas, P. Maes (1998) “Agent-mediated Electronic Commerce: A Survey”, Knowledge Engineering Review 13(3).

Huberman, B., S.H. Clearwater (1995) “A Multi-Agent System for Controlling Building Environments”, Proc. of the International Conference on Multiagent Systems (ICMAS ’95): 171-176.

Huberman, B., T. Hogg (1995) “Distributed computation as an economic system”, Journal of Economic Perspectives, 9(1):141-152.

JavaSoft Home Page (1997) Sun Microsystems, Inc., Mountain View, CA, USA, http://java.sun.com.

Java API for XML Parsing (JAXP) (1999) Sun Microsystems, Inc., Mountain View, CA, USA, http://java.sun.com/xml.

LeBaron, B. (2000) “Agent-based computational finance: Suggested readings and early research”, Journal of Economic Dynamics and Control 24:674-702.

Rodriquez, J., F. Martin, P. Noriega, P. Garcia, and C. Sierra (1998) “Competitive Scenarios for Heterogeneous Trading Agents”, Proceedings of the Second International Conference on Autonomous Agents (Agents’98).

Rust, J., J. Miller, R. Palmer, (1993) “Behavior of Trading Automata in a Computerized Double Auction Market”, in Friedman, D., J. Rust (eds.) “The Double Auction Market – Institutions, Theories, and Evidence”, Proceedings volume, Santa Fe Institute studies in sciences of complexity: 14, Addison-Wesley.

Rust, J., J. Miller, R. Palmer, (1994)“Characterizing effective trading strategies. Insight from a computerized double auction tournament”, Journal of Economic Dynamics and Control 18:61-96.

Selten, R., M. Mitzkewitz, G.R. Uhlich (1997) “Duoploy Strategies Programmed by Experienced Players”, Econometrica, 65(3):517-555.

Smith, V.L. (1976) “Experimental Economics: Induced Value Theory”, Amer. Econ. Rev. 66(2): 274-279.

Tsvetovatyy, M, M. Gini, B. Mosbasher, Z. Wieckowsky, (1997) “MAGMA: An Agent-based Virtual Market for Electronic Commerce”, Journal of Applied Artificial Intelligence, 6, Special Issue on Intelligent Agents.

Wurman, P.R., W.E. Walsh, M.P. Wellmann (1998) “Flexible Double Auctions for Electronic Commerce: Theory and Implementation”, Decision Support Systems 24:17-27.

Wurman, P.R., M.P. Wellmann and W.E. Walsh, (1998) “The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents”, Proceedings of the Second International Conference on Autonomous Agents.

Varian, H.R. (1995) “Economic mechanism design for computerized agents”, USENIX Workshop on Electronic Commerce, New York.